

Listing of the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A method for translating between an XML-type document and a first type of document in accordance with code residing in a storage device, comprising:

generating a data model for the XML-type document based on an XML data source;
generating a data model for the first type of document based on the XML data source;
creating mapping rules between the data model for the XML-type document and the data model for the first type of document;

storing the data model for the XML-type document, the data model for the first type of document, and the mapping rules in the storage device; and

verifying that the XML-type document is well-formed based upon the data model for the XML-type document.

2. (Original) A method according to claim 1, further comprising:

creating an executable file to effect the translation between the XML-type document and the first type of document based on the data model for the XML-type document, the data model for the first type of document and the mapping rules; and

running the executable file to translate between the XML-type document and the first type of document.

3. (Original) A method according to claim 1, further comprising:

translating test data based on the data model for the XML-type document, the data model for the first type of document and the mapping rules;

verifying the propriety of the data model for the XML-type document, the data model for the first type of document and the mapping rules based on the result of the translation.

4. (Original) A method according to claim 1, further comprising:

modifying the data model for the first type of document to conform to a format associated with the first type of document, and

modifying the mapping rules based on the modification of the data model for the first type of document.

5. **(Original)** A method according to claim 4, wherein the modifying the data model includes adjusting the data model for the first type of document to conform with an import utility of an application associated with the first type of document.

6. **(Original)** A method according to claim 1, wherein the generating of the data model for the first type of document includes omitting formatting that is present in the data model for the XML-type document.

7. **(Original)** A method according to claim 1, further comprising receiving an indication of the direction of the translation.

8. **(Original)** A method according to claim 1, further comprising receiving an indication of the identity of the XML data source.

9. **(Original)** A method according to claim 1, wherein the XML data source is an XML Schema Definition (XSD).

10. **(Original)** A method according to claim 9, wherein the generating of the data model for the XML-type document includes providing a model for numerics in the XSD.

11. **(Original)** A method according to claim 9, wherein the generating of the data model for the XML-type document includes providing a model for grouping and pattern definitions in the XSD.

12. **(Original)** A method according to claim 9, wherein the generating of the data model for the XML-type document includes providing a model for field lengths and value ranges.

13. (Original) A method according to claim 1, further comprising:
creating a map component file, which identifies the data model for the XML-type document, the data model for the first type of document,
wherein running the executable file includes referencing the map component file to perform the translation.

14. (Original) A method according to claim 1, further comprising receiving an indication of the identity of the XML-type document to be translated.

15. (Original) A method according to claim 1, further comprising validating that the translation between the XML-type document and the first type of document is accurate.

16. (Original) A method according to claim 15, further comprising receiving an indication of how to perform the validation.

17. (Canceled)

18. (Previously presented) A method according to claim 1, wherein the verifying includes determining that each element in the XML-type document has start and end tags with the same label.

19. (Canceled)

20. (Previously Presented) A method according to claim 1, wherein the verifying further comprises:

determining that elements in the XML-type document are in the correct order;
determining that the XML-type document includes any specified mandatory elements;
determining if data types in the XML-type document are proper; and
determining if the format of a value in a field in the XML-type document is proper.

21. (Previously Amended) A computer readable medium on a computer system for translating between an XML-type document and a first type of document, the computer readable medium configured to:

generate a data model for the XML-type document based on an XML data source;

generate a data model for the first type of document based on the XML data source;

create mapping rules between the data model for the XML-type document and the data model for the first type of document; and

verify that the XML-type document is well-formed based upon the data model for the XML-type document.

22. (Original) A computer readable medium according to claim 21, further configured to:

create an executable file to effect the translation between the XML-type document and the first type of document based on the data model for the XML-type document, the data model for the first type of document and the mapping rules; and

run the executable file to translate between the XML-type document and the first type of document.

23. (Original) A computer readable medium according to claim 21, further comprising:

translating test data based on the data model for the XML-type document, the data model for the first type of document and the mapping rules;

verifying the propriety of the data model for the XML-type document, the data model for the first type of document and the mapping rules based on the result of the translation.

24. (Original) A computer readable medium according to claim 21, further configured to:

modify the data model for the first type of document to conform to a format associated with the first type of document, and

modify the mapping rules based on the modification of the data model for the first type of document.

25. (Original) A computer readable medium according to claim 24, further configured to adjust the data model for the first type of document to conform with an import utility of an application associated with the first type of document.

26. (Original) A computer readable medium according to claim 21, further configured to omit formatting present in the data model for the XML-type document when generating the data model for the first type of document.

27. (Original) A computer readable medium according to claim 21, further configured to receive an indication of the direction of the translation.

28. (Original) A computer readable medium according to claim 21, further configured to receive an indication of the identity of the XML data source.

29. (Original) A computer readable medium according to claim 21, wherein the XML data source is an XML Schema Definition (XSD).

30. (Original) A computer readable medium according to claim 29, further configured to provide a model for numerics in the XSD.

31. (Original) A computer readable medium according to claim 29, further configured to provide a model for grouping and pattern definitions in the XSD.

32. (Original) A computer readable medium according to claim 29, further configured to provide a model for field lengths and value ranges.

33. (Original) A computer readable medium according to claim 21, further configured to:

create a map component file, which identifies the data model for the XML-type document, the data model for the first type of document; and

reference the map component file to perform the translation.

34. (Original) A computer readable medium according to claim 21, further configured to receive an indication of the identity of the XML-type document to be translated.

35. (Original) A computer readable medium according to claim 21, further configured to validate that the translation between the XML-type document and the first type of document is accurate.

36. (Original) A computer readable medium according to claim 35, further configured to receive an indication of how to perform the validation.

37. (Canceled)

38. (Previously Presented) A computer readable medium according to claim 21, further configured to determine that each element in the XML-type document has start and end tags with the same label.

39. (Canceled)

40. (Previously Presented) A computer readable medium according to claim 21, further configured to:

determine that elements in the XML-type document are in the correct order;

determine that the XML-type document includes any specified mandatory elements;

determine if data types in the XML-type document are proper; and

determine if the format of a value in a field in the XML-type document is proper.

41. (Previously Presented) A computer system for translating between an XML-type document and a first type of document, comprising:

a processor; and

a memory, coupled to the processor, comprising a plurality of instructions executed by the processor, the plurality of instructions configured to:

generate a data model for the XML-type document based on an XML data source;

generate a data model for the first type of document based on the XML data source;
create mapping rules between the data model for the XML-type document and the data model for the first type of document; and
verify that the XML-type document is well-formed based upon the data model for the XML-type document.

42. (Original) A computer system according to claim 41, further comprising instructions configured to:

create an executable file to effect the translation between the XML-type document and the first type of document based on the data model for the XML-type document, the data model for the first type of document and the mapping rules; and

run the executable file to translate between the XML-type document and the first type of document.

43. (Original) A computer system according to claim 41, further comprising instructions configured to:

modify the data model for the first type of document to conform to a format associated with the first type of document, and

modify the mapping rules based on the modification of the data model for the first type of document.

44. (Original) A computer system according to claim 41, further comprising an instruction configured to omit formatting present in the data model for the XML-type document when generating the data model for the first type of document.

45. (Previously Presented) A computer system for translating between an XML-type document and a first type of document, comprising:

means for generating a data model for the XML-type document based on an XML data source;

means for generating a data model for the first type of document based on the XML data source;

means for creating mapping rules between the data model for the XML-type document and the data model for the first type of document; and

means for verifying that the XML-type document is well-formed based upon the data model for the XML-type document.

46. (Original) A computer system according to claim 45, further comprising:

means for creating an executable file to effect the translation between the XML-type document and the first type of document based on the data model for the XML-type document, the data model for the first type of document and the mapping rules; and

means for running the executable file to translate between the XML-type document and the first type of document.

47. (Original) A computer system according to claim 45, further comprising:

means for modifying the data model for the first type of document to conform to a format associated with the first type of document, and

means for modifying the mapping rules based on the modification of the data model for the first type of document.

48. (Original) A computer system according to claim 45, further comprising

means for omitting formatting present in the data model for the XML-type document when generating the data model for the first type of document.